This listing of claims replaces all prior versions, and listings, of claims in this application.

Listing of Claims:

1. (Currently Amended) A method for certifying a software product's reliability, said method comprising ~~the steps of~~:

~~(a)~~ establishing a software certification laboratory;

~~(b)~~ receiving the software product at the software certification laboratory;

~~(c)~~ instrumenting the software product, by the software certification laboratory, to collect a plurality of usage data and a plurality of failure data when the software product is being executed;

~~(d)~~ providing the software product to a plurality of users;

~~(e)~~ receiving, at the software certification laboratory, the plurality of usage data and the plurality of failure data from each of the plurality of users;

~~(f)~~ analyzing the plurality of usage data and the plurality of failure data; ~~and~~ building an operational profile according to a result from the analyzing; and

~~(g)~~ issuing a certificate of reliability to the software product based on the result from the analyzing.

2. (New) The method of claim 1, wherein the software certification laboratory is independent of a vendor that provides the software product.

3. (New) The method of claim 1, wherein the operational profile is returned from the software certification laboratory to a vendor that provides the software product for reference.

4. (New) The method of claim 1, further comprising:

digitally signing the software product to be verified;

verifying the digital signature of the software product; and

employing a residual testing tool in the software product that collects information about the software product behavior as the software product is used.

5. (New) The method of claim 4, wherein the signature of the software product is signed by a software product vendor.

6. (New) The method of claim 4, wherein the signature of the software is verified by the software certification laboratory.

7. (New) The method of claim 1, wherein the certification is presented by computing a Mean Time to Failure (MTTF) for the software product, wherein the MTTF is given by

$$\text{MTTF} = \frac{\sum_{i=1}^{N} t_i}{N}$$, wherein N is a number of instrumented versions of the software product, and t_i represents the time for each version before encountering a failure.

8. (New) A system for certifying a software product's reliability, comprising:

a software product received from a product vendor; and

a software certification laboratory for instrumenting the software product to collect information about the software behavior when the software product is used,

wherein the software product, after intrumented by the software certification laboratory, is used by a plurality of users, and the information collected by the software products includes a plurality of usage data and a plurality of failure data from each of the plurality of users,

wherein the software certification laboratory receives and analyzes the plurality of usage data and the plurality of failure data from each of the plurality of users collected by the software product, builds an operational profile according to a result from the analyzing, and issues a certificate of reliability to the software product.

9. (New) The system of claim 8, wherein the software certification laboratory instruments the software product with a residual testing tool that collects information about the software behavior when the software product is used.

10. (New) The system of claim 8, wherein the software product is provided by a software vendor that is independent from the software certification laboratory.

11. (New) The system of claim 10, wherein the vendor provides a digital signature on the software product.

12. (New) The system of claim 11, wherein the software certification laboratory verifies the digital signature of the software product.

13. (New) The system of claim 8, wherein the operational profile is returned from the software certification laboratory to a vendor that provides the software product for reference.

14. (New) The system of claim 8, wherein the collected data is scrubbed of personal information of the plurality of users.

15. (New) The system of claim 8, wherein the software certification laboratory instructs the software product to collect the information periodically.

16. (New) The system of claim 8, wherein the certification is presented by computing a Mean Time to Failure (MTTF) for the software product, wherein the MTTF is given by

$\text{MTTF} = \frac{\sum_{i=1}^{N} t_i}{N}$, wherein N is the number of instrumented versions of the software product, and t_i represents the time for each version before encountering a failure.

17. (New) A method for certifying a software product's reliability, comprising:

employing a residual testing tool to the software, wherein the residual testing tool is configured to collects information about the software product behavior as the software product is used;

digitally signing the software product;

distributing the signed software product to a software certification laboratory, wherein the software certification laboratory verifies the digital signature of the software product and instruments the residual testing tool employed in the software product to collect information about the software product behavior;

distributing the verified software product to a plurality of users, wherein the residual testing tool collects a plurality of usage data and a plurality of failure data from each of the plurality of users while the software product is used;

reporting the collected usage and failure data to the software certification laboratory; and

analyzing the collected data to issues a certificate of reliability to the software product.

18. (New) The method of claim 17, wherein the software certification laboratory further builds an operational profile according to a result from the analyzing.

19. (New) The method of claim 17, wherein the certification is presented by computing a Mean Time to Failure (MTTF) for the software product, wherein the MTTF is given by

$$\text{MTTF} = \frac{\sum_{i=1}^{N} t_i}{N}$$, wherein N is a number of instrumented versions of the software product, and t_i represents the time for each version before encountering a failure.